Detailed Description Text - DETX (21):

After the row address has been applied to the address register 182 and stored in one of the row address latches 200, a column address is applied to the address register 182. The address register 182 couples the column address to a column address latch 220. Depending on the operating mode of the SDRAM 180, the column address is either coupled through a burst counter 222 to a column address register 224 or to the burst counter 222 which applies a sequence of column addresses to the column address buffer 224 starting at the column address output by the address register 182. In either case, the column address buffer 224 applies a column address to a column decoder 228 which applies various column signals to respective sense amplifiers and associated circuitry 230, 232 for the respective arrays 190, 192.

Detailed Description Text - DETX (23):

The above-described operation of the SDRAM 180 is controlled by the Command Generator 26 responsive to high level command signals received on a control bus 160. These high level command signals, which are typically generated by a memory controller (not shown in FIG. 10), are a clock enable signal CKE*, a clock signal CLK, a chip select signal CS*, a write enable signal WE*, a row address strobe signal RAS*, and a column address strobe signal CAS*, which the "*" designating the signal as active low. However, other high level command signals may be used. In either case, the Command Generator 26 generates a sequence of command signals responsive to the high level command signals to carry out the function (e.g., a read or a write) designated by each of the high level command signals. These command signals, and the manner in which they accomplish their respective functions, are conventional. Therefore, in the interest of brevity, a further explanation of these control signals will be omitted.

Detailed Description Text - DETX (24):

Although the Command Generator 26 has been described as generating command signals for an SDRAM, it will be understood that a sequence of command signals for other varieties of DRAMs, as well as other integrated circuit devices, may be generated in a similar manner.

Detailed Description Text - DETX (25):

FIG. 11 is a block diagram of a computer system 300 which includes the SDRAM 180 of FIG. 10. The computer system 300 includes a processor 302 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 302 includes a processor bus 304 that normally includes an address bus, a control bus, and a data bus. In addition, the computer system 300 includes one or more input devices 314, such as a keyboard or a mouse, coupled to the processor 302 to allow an operator to interface with the computer system 300. Typically, the computer system 300 also includes one or more output devices 316 coupled to the processor 302, such output devices typically being a printer or a video terminal. One or more data storage devices 318 are also typically coupled to the processor 302 to store data or retrieve data from external storage media (not shown). Examples of typical storage devices 318 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs). The processor 302 is also typically coupled to cache memory 326, which is usually static random access memory ("SRAM") and to the **SDRAM** 180 through a memory controller 330. The memory controller 330 normally includes the control bus 160 and the address bus 184 that is coupled to the **SDRAM** 180. The data bus 242 may be coupled to the processor bus 304 either directly (as shown), through the memory controller 330, or by some other means.

US-PAT-NO: 6230245

DOCUMENT-IDENTIFIER: US 6230245 B1
See image for Certificate of Correction

TITLE: Method and apparatus for generating a variable sequence of memory device command signals

DATE-ISSUED: May 8, 2001

US-CL-CURRENT: 711/167, 365/230.01 , 365/230.05 , 711/105 , 711/154

APPL-NO: 08/ 798229

DATE FILED: February 11, 1997

——— KWIC ———

Abstract Text - ABTX (1):

A command generator for a dynamic random access memory decrements a counter from an initial counter value which is a function of the clock speed. The output of the counter is decoded to generate various command signals for the DRAM. In particular, each command signal is generated by a respective counter value, with the correspondency between counter values and command signals being a function of the clock speed. The counter decrements from larger initial values at higher clock speeds, and the command signals are generally issued by the decoder at higher counter values for higher clock speeds. As a result of the lack of correspondency between the timing of the command signals and the number of clock cycles occurring during a memory access, the timing of the command signals may be selected to optimize the speed of the DRAM desired despite substantial variations in clock speed.

Brief Summary Text - BSTX (8):

A command generator for generating command signals for a memory device includes a sequencer generating a sequence of command signals responsive to a clock signal that may have one of a plurality of clock speeds. Each of the sequences of command signals preferably corresponds to a respective clock speed, and the sequencer selects one of the sequences as a function of the clock speed. The sequencer may include a counter and a decoder. The counter receives the clock signal and provides a counter value that increments or decrements responsive to the clock signal. The decoder generates one of a plurality of sequences of command signals, with the command signals in each sequence corresponding to respective counter values. Each of the sequences of command signals corresponds to a respective clock speed, and the decoder selects one of the sequences as a function of the clock speed. Thus, the correspondency between each command signal and its respective counter value is a function of the clock speed. The command generator may also include a counter load circuit coupled to the counter. The counter load circuit loads an

US-PAT-NO: 4963866

DOCUMENT-IDENTIFIER: US 4963866 A

TITLE: Multi channel digital random access recorder-player

DATE-ISSUED: October 16, 1990

US-CL-CURRENT: 341/110, 341/123 , 360/32 , 365/45

APPL-NO: 07/ 328863

DATE FILED: March 27, 1989

——— KWIC ———

Detailed Description Text - DETX (57):

When turned on, memory manager U22 pulls the CE bar line low enabling data to be written to the memory and resetting the counters. As the message is recorded, cascaded address counters U13, U14, U15, and U16 count the pulses of the counter strobe and use the count to provide addressing for memory. (Organized as 64k addresses with 1 bit of data per address). During a record cycle, all ones are written into memory. When finished recording, the user switches from record to play. The record/play line goes low, and the system keeps counting by writing in zeros (the stop code) into memory. A "D" flip flop logic circuit allows the address counters to increment two more times, then pulses the last board out line which resets the recorder and clears the address counters. This is to insure separation between the old and new messages.

L Number	Hits	Search Text	DB	Time stamp
-	2332	strob$4 near3 count$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:51
-	510	(strob$4 near3 count$4) same enabl$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/12 15:02
-	207	((strob$4 near3 count$4) same enabl$4) same reset$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/12 14:52
-	156	(strob$4 near3 count$4) same (enabl$4 or start$4) same (disabl$4 or stop$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:52
-	2332	strob$4 near3 count$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:51
-	79773	(clock or edge) near3 count$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:52
-	80979	(strob$4 near3 count$4) or ((clock or edge) near3 count$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:52
-	6704	((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:59
-	25731	191.ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:53
-	52881	(((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4)) ans 191.ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:53
-	1369	(((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4)) and 191.ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 16:18
-	25731	((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) .ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:53
-	3721	((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) .ti.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:54
-	1535	(((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) .ti.) and 191.ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:54

-	65	(((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) .ti.) and (((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:54
-	1056	(((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4)) same memory	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:59
-	132	((((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4)) same memory) same controller	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 15:59
-	13821	((((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4)) and 191.ab.) ddr	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 16:18
-	1	((((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4)) and 191.ab.) and ddr	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 16:18
-	0	((((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4)) and 191.ab.) and rambus	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 16:18
-	4	((((strob$4 near3 count$4) or ((clock or edge) near3 count$4)) same (enabl$4 or start$4) same (disabl$4 or stop$4)) and 191.ab.) and sdram	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 17:10
-	2	("20020172079").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/14 17:11

L Number	Hits	Search Text	DB	Time stamp
-	12940	ddr or rambus	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 14:08
-	561	(ddr or rambus) same controller	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 14:16
-	12	((ddr or rambus) same controller) same noise	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 14:16
-	68728	memory near3 controller	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 14:16
-	558	(memory near3 controller) same noise	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 14:17
-	31	((memory near3 controller) same noise) same strob$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 14:44
-	26	(((memory near3 controller) same noise) same strob$4) and counter	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 14:43
-	3	(((memory near3 controller) same noise) same strob$4) same counter	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 14:43
-	8278	counter same strob$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 15:58
-	2667	(counter same strob$4) same memory	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 14:44
-	538	((counter same strob$4) same memory) and feedback	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 15:59
-	301	(((counter same strob$4) same memory) and feedback) and controller	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 15:59
-	26	((counter same strob$4) same memory) same feedback	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/23 17:08
-	344	((counter same strob$4) same memory) same controller	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 16:02

-	73	(((counter same strob$4) same memory) same controller) and feedback	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/12/02 16:02
-	27	((counter same strob$4) same memory) same feedback	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/23 17:08
-	27	(("6532525") or ("6530001") or ("20020172079") or ("20020147898") or ("20020147896") or ("20020147892") or ("6414868") or ("6370630") or ("20010046163") or ("6125078") or ("6065132") or ("5809340") or ("5727005") or ("5522064")).PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 17:52
-	56851	strob$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:10
-	9894	strob$4 same count$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:11
-	184234	clock$4 same count$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:12
-	198	(strob$4 same count$4) same asynchronous$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:12
-	2810	(clock$4 same count$4) same asynchronous$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:12
-	749	((clock$4 same count$4) same asynchronous$4) same reset$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:13
-	67	((strob$4 same count$4) same asynchronous$4) same reset$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:21
-	1082800	count$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:22
-	11763	count$4 near3 edg$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 00:08
-	2195	(count$4 near3 edg$4) same reset$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:25
-	243	121.ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:24

-	15	121.ti.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:24
-	512	((count$4 near3 edg$4) same reset$4) same enabl$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 00:24
-	76	(((count$4 near3 edg$4) same reset$4) same memory) and (((count$4 near3 edg$4) same reset$4) same enabl$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 00:05
-	20	156.ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:31
-	2857	107.ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:38
-	27	(((count$4 near3 edg$4) same reset$4) same enabl$4) and 107.ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:38
-	230	((count$4 near3 edg$4) same reset$4) same memory	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:44
-	952	(count$4 near3 edg$4) same memory	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:55
-	145	((count$4 near3 edg$4) same memory) same enabl$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:55
-	60	(((count$4 near3 edg$4) same memory) same enabl$4) same reset$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/26 22:56
-	60	(((count$4 near3 edg$4) same reset$4) same enabl$4) same memory	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 00:06
-	5	(count$4 near3 edg$4) same ddr	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 00:08
-	147	(count$4 near3 edg$4) same strobe	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 00:13
-	45	((count$4 near3 edg$4) same strobe) same enabl$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 00:25

-	27	(("6532525") or ("6530001") or ("20020172079") or ("20020147898") or ("20020147896") or ("20020147892") or ("6414868") or ("6370630") or ("20010046163") or ("6125078") or ("6065132") or ("5809340") or ("5727005") or ("5522064")).PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 17:52
-	12	((("6532525") or ("6530001") or ("20020172079") or ("20020147898") or ("20020147896") or ("20020147892") or ("6414868") or ("6370630") or ("20010046163") or ("6125078") or ("6065132") or ("5809340") or ("5727005") or ("5522064")).PN.) and counter	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/27 17:53

PGPUB-DOCUMENT-NUMBER: 20030070037

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030070037 A1

TITLE: Memory device command signal generator

PUBLICATION-DATE: April 10, 2003

US-CL-CURRENT: 711/105, 711/167

APPL-NO: 09/ 812622

DATE FILED: March 19, 2001

RELATED-US-APPL-DATA:

child 09812622 A1 20010319

parent continuation-of 08798229 19970211 US GRANTED

parent-patent 6230245 US

——— KWIC ———

Abstract Paragraph - ABTX (1):

A command generator for a dynamic random access memory decrements a counter from an initial counter value which is a function of the clock speed. The output of the counter is decoded to generate various command signals for the DRAM. In particular, each command signal is generated by a respective counter value, with the correspondency between counter values and command signals being a function of the clock speed. The counter decrements from larger initial values at higher clock speeds, and the command signals are generally issued by the decoder at higher counter values for higher clock speeds. As a result of the lack of correspondency between the timing of the command signals and the number of clock cycles occurring during a memory access, the timing of the command signals may be selected to optimize the speed of the DRAM desired despite substantial variations in clock speed.

Summary of Invention Paragraph - BSTX (8):

[0005] A command generator for generating command signals for a memory device includes a sequencer generating a sequence of command signals responsive to a clock signal that may have one of a plurality of clock speeds. Each of the sequences of command signals preferably corresponds to a respective clock speed, and the sequencer selects one of the sequences as a function of the clock speed. The sequencer may include a counter and a decoder. The counter receives the clock signal and provides a counter value that increments or decrements responsive to the clock signal. The decoder generates one of a plurality of sequences of command signals, with the command signals in each sequence corresponding to respective counter values. Each of the sequences of command signals corresponds to a respective clock speed, and the decoder selects one of the sequences as a function of the clock speed. Thus, the correspondency between each command signal and its respective counter value is a function of the clock speed. The command generator may also include a counter load circuit coupled to the counter. The counter load circuit loads an initial count into the counter that is a function of the clock speed. The counter then increments or decrements from the initial value responsive to the clock signal. The command generator may also include a counter enable circuit generating a counter enable signal to permit the counter to increment or decrement responsive to the clock signal. The counter enable circuit MAY include a latch circuit and a counter start circuit. The latch circuit generates the counter enable signal responsive to a start signal and terminates the counter enable signal responsive to a stop signal. The counter start circuit generates the start signal and includes a clock detector detecting predetermined portions of the clock signal, and a variable delay enable circuit coupled to the clock detector. The variable delay enable circuit generates the start signal responsive to one of the detected predetermined portions of the clock signal after a predetermined number of cycles of the clock signal have elapsed from receiving a flag signal. The command generator may be used in any type of dynamic random access memory or other circuit which may be part of a computer system.

Detail Description Paragraph - DETX (11):

[0026] The operation of the Command Generator 26 illustrated in FIG. 3 is best explained with reference to the timing diagram of FIG. 5. The speed signal SPEED corresponding to the clock speed is output to the Counter Start Logic circuit 40 and the Decoder 56, as explained above. When the FLAG signal is received by the shift register 34, the shift register 34 generates a sequence of F signals, one of which is used by the Counter Start Logic circuit 40 to generate the START signal. However, as illustrated in FIG. 5, prior to the START signal, a LOAD signal loads the initial counter value LD CNT from the Load Register 52, and the STOP signal goes inactive low. In response to the positive going START signal illustrated in FIG. 5, the Counter Control circuit 46 enables the Counter 50 so that it decrements from the initial counter value. The Decoder 56 then generates appropriate command signals responsive to the counter values and the SPEED signal. At or before the terminal count, other circuitry in the integrated circuit causes the STOP signal to go active high, thereby causing the Counter Control circuit 46 to disable the Counter 50.

Detail Description Paragraph - DETX (20):

[0035] A synchronous DRAM ("SDRAM") 180 using the Command Generator 26 of FIG. 3 is shown in FIG. 10. The SDRAM 180 includes an address register 182 that receives either a row address or a column address on an address bus 184. The address bus 184 is generally coupled to a memory controller (not shown in FIG. 10). A row address is initially received by the address register 182 and applied to a row address multiplexer 188. The row address multiplexer 188 couples the row address to a number of components associated with either of two memory banks 190, 192 depending upon the state of a bank address bit BA forming part of the row address. Associated with each of the memory banks 190, 192 are a respective row address latch 200 which stores the row address, and a row decoder 202 which applies various row signals to its respective array 190 or 192 as a function of the stored row address. The row address multiplexer 188 also couples row addresses to the row address latches 200 for the purpose of refreshing the memory cells in the arrays 190, 192. The row addresses are generated for refresh purposes by a refresh counter 210 which is controlled by a refresh controller 212.

Detail Description Paragraph - DETX (21):
[0036] After the row address has been applied to the address register 182 and stored in one of the row address latches 200, a column address is applied to the address register 182. The address register 182 couples the column address to a column address latch 220. Depending on the operating mode of the SDRAM 180, the column address is either coupled through a burst counter 222 to a column address register 224 or to the burst counter 222 which applies a sequence of column addresses to the column address buffer 224 starting at the column address output by the address register 182. In either case, the column address buffer 224 applies a column address to a column decoder 228 which applies various column signals to respective sense amplifiers and associated circuitry 230, 232 for the respective arrays 190, 192.

Detail Description Paragraph - DETX (23):
[0038] The above-described operation of the SDRAM 180 is controlled by the Command Generator 26 responsive to high level command signals received on a control bus 160. These high level command signals, which are typically generated by a memory controller (not shown in FIG. 10), are a clock enable signal CKE*, a clock signal CLK, a chip select signal CS*, a write enable signal WE*, a row address strobe signal RAS*, and a column address strobe signal CAS*, which the "*" designating the signal as active low. However, other high level command signals may be used. In either case, the Command Generator 26 generates a sequence of command signals responsive to the high level command signals to carry out the function (e.g., a read or a write) designated by each of the high level command signals. These command signals, and the manner in which they accomplish their respective functions, are conventional. Therefore, in the interest of brevity, a further explanation of these control signals will be omitted.

Detail Description Paragraph - DETX (24):
[0039] Although the Command Generator 26 has been described as generating command signals for an SDRAM, it will be understood that a sequence of command signals for other varieties of DRAMs, as well as other integrated circuit devices, may be generated in a similar manner.

Detail Description Paragraph - DETX (25):
[0040] FIG. 11 is a block diagram of a computer system 300 which includes the SDRAM 180 of FIG. 10. The computer system 300 includes a processor 302 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 302 includes a processor bus 304 that normally includes an address bus, a control bus, and a data bus. In addition, the computer system 300 includes one or more input devices 314, such as a keyboard or a mouse, coupled to the processor 302 to allow an

operator to interface with the computer system 300. Typically, the computer system 300 also includes one or more output devices 316 coupled to the processor 302, such output devices typically being a printer or a video terminal. One or more data storage devices 318 are also typically coupled to the processor 302 to store data or retrieve data from external storage media (not shown). Examples of typical storage devices 318 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs). The processor 302 is also typically coupled to cache memory 326, which is usually static random access memory ("SRAM") and to the SDRAM 180 through a memory controller 330. The memory controller 330 normally includes the control bus 160 and the address bus 184 that is coupled to the SDRAM 180. The data bus 242 may be coupled to the processor bus 304 either directly (as shown), through the memory controller 330, or by some other means.

initial count into the counter that is a function of the clock speed. The counter then increments or decrements from the initial value responsive to the clock signal. The command generator may also include a counter enable circuit generating a counter enable signal to permit the counter to increment or decrement responsive to the clock signal. The counter enable circuit MAY include a latch circuit and a counter start circuit. The latch circuit generates the counter enable signal responsive to a start signal and terminates the counter enable signal responsive to a stop signal. The counter start circuit generates the start signal and includes a clock detector detecting predetermined portions of the clock signal, and a variable delay enable circuit coupled to the clock detector. The variable delay enable circuit generates the start signal responsive to one of the detected predetermined portions of the clock signal after a predetermined number of cycles of the clock signal have elapsed from receiving a flag signal. The command generator may be used in any type of dynamic random access memory or other circuit which may be part of a computer system.

Detailed Description Text - DETX (11):

The operation of the Command Generator 26 illustrated in FIG. 3 is best explained with reference to the timing diagram of FIG. 5. The speed signal SPEED corresponding to the clock speed is output to the Counter Start Logic circuit 40 and the Decoder 56, as explained above. When the FLAG signal is received by the shift register 34, the shift register 34 generates a sequence of F signals, one of which is used by the Counter Start Logic circuit 40 to generate the START signal. However, as illustrated in FIG. 5, prior to the START signal, a LOAD signal loads the initial counter value LD CNT from the Load Register 52, and the STOP signal goes inactive low. In response to the positive going START signal illustrated in FIG. 5, the Counter Control circuit 46 enables the Counter 50 so that it decrements from the initial counter value. The Decoder 56 then generates appropriate command signals responsive to the counter values and the SPEED signal. At or before the terminal count, other circuitry in the integrated circuit causes the STOP signal to go active high, thereby causing the Counter Control circuit 46 to disable the Counter 50.

Detailed Description Text - DETX (20):

A synchronous DRAM ("SDRAM") 180 using the Command Generator 26 of FIG. 3 is shown in FIG. 10. The SDRAM 180 includes an address register 182 that receives either a row address or a column address on an address bus 184. The address bus 184 is generally coupled to a memory controller (not shown in FIG. 10). A row address is initially received by the address register 182 and applied to a row address multiplexer 188. The row address multiplexer 188 couples the row address to a number of components associated with either of two memory banks 190, 192 depending upon the state of a bank address bit BA forming part of the row address. Associated with each of the memory banks 190, 192 are a respective row address latch 200 which stores the row address, and a row decoder 202 which applies various row signals to its respective array 190 or 192 as a function of the stored row address. The row address multiplexer 188 also couples row addresses to the row address latches 200 for the purpose of refreshing the memory cells in the arrays 190, 192. The row addresses are generated for refresh purposes by a refresh counter 210 which is controlled by a refresh controller 212.